Listing of the Claims

The following list, if entered, replaces all prior versions of the claims.

1. **(Currently Amended)** A method comprising:

selecting a first allocated memory block from a plurality of allocated memory blocks, wherein the first allocated memory block includes a first allocated memory block address;

searching other allocated memory blocks of the plurality of allocated memory blocks for a reference to the first allocated memory block;

identifying that the first allocated memory block is a suspected memory leak when the reference to the first allocated memory block is not found in the other allocated memory blocks of the plurality of allocated memory blocks;

verifying that the first allocated memory block is a memory leak in response to the first allocated memory block being identified as **[[a]]** **the** suspected memory leak; and

reporting the first allocated memory block as **[[a]]** **the** memory leak when the verifying confirms that the first allocated memory block is **[[a]]** **the** memory leak.

2. (Original) The method of claim 1 wherein the selecting the first allocated memory block from the plurality of allocated memory blocks further comprises:

selecting the first allocated memory block address from operating system memory management information.

3. **(Currently Amended)** The method of claim 1 wherein each of the plurality of allocated memory blocks includes a header portion, and wherein the searching other allocated memory blocks of the plurality of allocated memory blocks further comprises:

searching the header portions of the other allocated memory blocks of the plurality of allocated memory blocks for **[[a]]** **the** reference to the first allocated memory block.

4. (Original) The method of claim 1 wherein the searching other allocated memory blocks of the plurality of allocated memory blocks further comprises:

searching for an occurrence of the first allocated memory block address in the other allocated memory blocks of the plurality of allocated memory blocks.

5. (Original) The method of claim 1 further comprising:

examining a reference counter corresponding to the first allocated memory block.

6. (Original) The method of claim 1 wherein the verifying that the first allocated memory block is a memory leak further comprises:

determining whether the first allocated memory block has been deallocated.

7. (Original) The method of claim 1 wherein the first allocated memory block includes a header portion, and wherein the verifying that the first allocated memory block is a memory leak further comprises:

examining the header portion of the first allocated memory block.

8. (Original) The method of claim 1 wherein the verifying that the first allocated memory block is a memory leak further comprises:

examining free block memory management information maintained by an operating system.

9. **(Currently Amended)** The method of claim 1 wherein the reporting the first allocated memory block as **[[a]]** **the** memory leak further comprises:

displaying to a user at least one of: a program counter value, a process identification value, a process name, an initial block count, a previous block count, a current block count, a linearity value, the first allocated memory block address, and contents of the first allocated memory block.

10. (Original) The method of claim 1 wherein the reporting the first allocated memory block as a memory leak further comprises:

storing in a data structure at least one of: a program counter value, a process identification value, a process name, an initial block count, a previous block count, a current block count, a linearity value, the first allocated memory block address, and contents of the first allocated memory block.

11. **(Currently Amended)** The method of claim 1 further comprising:

searching the first allocated memory block for a reference to at least one of the plurality **of** allocated memory blocks; and

storing the first allocated memory block address in a contingency chain corresponding to the at least one of the plurality **of** allocated memory blocks when a reference to the at least one of the plurality **of** allocated memory blocks is found in the first allocated memory block.

12. **(Currently Amended)** The method of claim 1 further comprising:

examining a contingency chain corresponding to one of the plurality of allocated memory blocks to determine whether any of the plurality of allocated memory blocks references the one of the plurality of allocated memory **blocks**.

13. (Original) The method of claim 1 further comprising:

forming a contingency chain for each of the plurality of allocated memory blocks, wherein each contingency chain is indexed by an allocated memory block address of the corresponding each of the plurality of allocated memory blocks.

14. **(Currently Amended)** A system comprising:

a memory;

a processor coupled to the memory; and

a memory leak detection system (MLDS) engine, wherein at least a portion of the MLDS engine is encoded as instructions stored in the memory and executable on the processor, and wherein the MLDS engine is configured to:

select a first allocated memory block from a plurality of allocated memory blocks stored in the memory, wherein the first allocated memory block includes a first allocated memory block address;

search other allocated memory blocks of the plurality of allocated memory blocks for a reference to the first allocated memory block;

identify that the first allocated memory block is a suspected memory leak when the reference to the first allocated memory block is not found in the other allocated memory blocks of the plurality of allocated memory blocks;

verify that the first allocated memory block is a memory leak in response to the first allocated memory block being identified as **[[a]]** **the** suspected memory leak; and

report the first allocated memory block as **[[a]]** **the** memory leak when the verifying confirms that the first allocated memory block is **[[a]]** **the** memory leak.

15. (Previously Presented) The system of claim 14 further comprising at least one of an MLDS data structure application programming interface (API), an MLDS command API, an MLDS data structure, and a command line interface (CLI) parser stored in at least one of the memory and a storage device accessible by the processor.

16. (Original) The system of claim 14 wherein the MLDS engine is further configured to:

select the first allocated memory block address from operating system memory management information.

17. **(Currently Amended)** The system of claim 14 wherein each of the plurality of allocated memory blocks includes a header portion, and wherein the MLDS engine is further configured to:

search the header portions of the other allocated memory blocks of the plurality of allocated memory blocks for **[[a]]** **the** reference to the first allocated memory block.

18. (Original) The system of claim 14 wherein the MLDS engine is further configured to:

search for an occurrence of the first allocated memory block address in the other allocated memory blocks of the plurality of allocated memory blocks.

19. (Original) The system of claim 14 wherein the MLDS engine is further configured to:

examine a reference counter corresponding to the first allocated memory block.

20. (Original) The system of claim 14 wherein the MLDS engine is further configured to

determine whether the first allocated memory block has been deallocated.

21. (Original) The system of claim 14 wherein the first allocated memory block includes a header portion, and wherein the MLDS engine is further configured to:

examine the header portion of the first allocated memory block.

22. (Original) The system of claim 14 wherein the MLDS engine is further configured to:

examine free block memory management information maintained by an operating system.

23. (Original) The system of claim 14 wherein the MLDS engine is further configured to:

display at least one of: a program counter value, a process identification value, a process name, an initial block count, a previous block count, a current block count, a linearity value, the first allocated memory block address, and contents of the first allocated memory block.

24. (Original) The system of claim 14 wherein the MLDS engine is further configured to:

store in a data structure at least one of: a program counter value, a process identification value, a process name, an initial block count, a previous block count, a current block count, a linearity value, the first allocated memory block address, and contents of the first allocated memory block.

25. **(Currently Amended)** The system of claim 14 wherein the MLDS engine is further configured to:

search the first allocated memory block for a reference to at least one of the plurality **of** allocated memory blocks; and

store the first allocated memory block address in a contingency chain corresponding to the at least one of the plurality **of** allocated memory blocks when a reference to the at least one of the plurality **of** allocated memory blocks is found in the first allocated memory block.

26. **(Currently Amended)** The system of claim 14 wherein the MLDS engine is further configured to:

examine a contingency chain corresponding to one of the plurality of allocated memory blocks to determine whether any of the plurality of allocated memory blocks references the one of the plurality of allocated memory **blocks**.

27. (Original) The system of claim 14 wherein the MLDS engine is further configured to:

form a contingency chain for each of the plurality of allocated memory blocks, wherein each contingency chain is indexed by an allocated memory block address of the corresponding each of the plurality of allocated memory blocks.

28. **(Currently Amended)** A computer readable storage medium comprising program instructions executable on a processor, wherein the program instructions are operable to implement each of:

selecting a first allocated memory block from a plurality of allocated memory blocks, wherein the first allocated memory block includes a first allocated memory block address;

searching other allocated memory blocks of the plurality of allocated memory blocks for a reference to the first allocated memory block;

identifying that the first allocated memory block is a suspected memory leak when the reference to the first allocated memory block is not found in the other allocated memory blocks of the plurality of allocated memory blocks;

verifying that the first allocated memory block is a memory leak in response to the first allocated memory block being identified as **[[a]]** **the** suspected memory leak; and

reporting the first allocated memory block as **[[a]]** **the** memory leak when the verifying confirms that the first allocated memory block is **[[a]]** **the** memory leak.

29. (Previously Presented) The computer readable storage medium of claim 28 wherein the selecting the first allocated memory block from the plurality of allocated memory blocks further comprises:

selecting the first allocated memory block address from operating system memory management information.

30. **(Currently Amended)** The computer readable storage medium of claim 28 wherein each of the plurality of allocated memory blocks includes a header portion, and wherein the searching other allocated memory blocks of the plurality of allocated memory blocks further comprises:

searching the header portions of the other allocated memory blocks of the plurality of allocated memory blocks for **[[a]]** **the** reference to the first allocated memory block.

31. (Previously Presented) The computer readable storage medium of claim 28 wherein the searching other allocated memory blocks of the plurality of allocated memory blocks further comprises:

searching for an occurrence of the first allocated memory block address in the other allocated memory blocks of the plurality of allocated memory blocks.

32. **(Currently Amended)** The computer readable storage medium of claim 28 further comprising program instructions **[[are]]** operable to implement:

examining a reference counter corresponding to the first allocated memory block.

33. **(Currently Amended)** The computer readable storage medium of claim 28 wherein the verifying that the first allocated memory block is **[[a]]** **the** memory leak further comprises:

determining whether the first allocated memory block has been deallocated.

34. **(Currently Amended)** The computer readable storage medium of claim 28 wherein the first allocated memory block includes a header portion, and wherein the verifying that the first allocated memory block is **[[a]]** **the** memory leak further comprises:

examining the header portion of the first allocated memory block.

35. **(Currently Amended)** The computer readable storage medium of claim 28 wherein the verifying that the first allocated memory block is **[[a]]** **the** memory leak further comprises:

examining free block memory management information maintained by an operating system.

36. (Previously Presented) The computer readable storage medium of claim 28 wherein the reporting the first allocated memory block as a memory leak further comprises:

displaying to a user at least one of: a program counter value, a process identification value, a process name, an initial block count, a previous block count, a current block count, a linearity value, the first allocated memory block address, and contents of the first allocated memory block.

37. (Previously Presented) The computer readable storage medium of claim 28 wherein the reporting the first allocated memory block as a memory leak further comprises:

storing in a data structure at least one of: a program counter value, a process identification value, a process name, an initial block count, a previous block count, a current block count, a linearity value, the first allocated memory block address, and contents of the first allocated memory block.

38. **(Currently Amended)** The computer readable storage medium of claim 28 further comprising program instructions **[[are]]** operable to implement each of:

searching the first allocated memory block for a reference to at least one of the plurality **of** allocated memory blocks; and

storing the first allocated memory block address in a contingency chain corresponding to the at least one of the plurality **of** allocated memory blocks when a reference to the at least one of the plurality **of** allocated memory blocks is found in the first allocated memory block.

39. **(Currently Amended)** The computer readable storage medium of claim 28 further comprising program instructions **[[are]]** operable to implement each of:

examining a contingency chain corresponding to one of the plurality of allocated memory blocks to determine whether any of the plurality of allocated memory blocks references the one of the plurality of allocated memory **blocks**.

40. **(Currently Amended)** The computer readable storage medium of claim 28 further comprising program instructions **[[are]]** operable to implement:

forming a contingency chain for each of the plurality of allocated memory blocks, wherein each contingency chain is indexed by an allocated memory block address of the corresponding each of the plurality of allocated memory blocks.

41. **(Currently Amended)** An apparatus comprising:

a means for selecting a first allocated memory block from a plurality of allocated memory blocks, wherein the first allocated memory block includes a first allocated memory block address;

a means for searching other allocated memory blocks of the plurality of allocated memory blocks for a reference to the first allocated memory block;

a means for identifying that the first allocated memory block is a suspected memory leak when the reference to the first allocated memory block is not found in the other allocated memory blocks of the plurality of allocated memory blocks;

a means for verifying that the first allocated memory block is a memory leak in response to the first allocated memory block being identified as **[[a]]** **the** suspected memory leak; and

a means for reporting the first allocated memory block as **[[a]]** **the** memory leak when the verifying confirms that the first allocated memory block is **[[a]]** **the** memory leak.

42. **(Currently Amended)** The apparatus of claim 41 wherein each of the plurality of allocated memory blocks includes a header portion, and wherein the apparatus further comprises:

a means for searching the header portions of the other allocated memory blocks of the plurality of allocated memory blocks for **[[a]]** **the** reference to the first allocated memory block.

43. (Original) The apparatus of claim 41 further comprising:

a means for searching for an occurrence of the first allocated memory block address in the other allocated memory blocks of the plurality of allocated memory blocks.

44. (Original) The apparatus of claim 41 further comprising:

a means for displaying to a user at least one of: a program counter value, a process identification value, a process name, an initial block count, a previous block count, a current block count, a linearity value, the first allocated memory block address, and contents of the first allocated memory block.

45. **(Currently Amended)** The apparatus of claim 41 further comprising:

a means for searching the first allocated memory block for a reference to at least one of the plurality **of** allocated memory blocks; and

a means for storing the first allocated memory block address in a contingency chain corresponding to the at least one of the plurality **of** allocated memory blocks when a reference to the at least one of the plurality **of** allocated memory blocks is found in the first allocated memory block.

46. **(Currently Amended)** The apparatus of claim 41 further comprising:

a means for examining a contingency chain corresponding to one of the plurality of allocated memory blocks to determine whether any of the plurality of allocated memory blocks references the one of the plurality of allocated memory **blocks**.

47. (Original) The apparatus of claim 41 further comprising:

a means for forming a contingency chain for each of the plurality of allocated memory blocks, wherein each contingency chain is indexed by an allocated memory block address of the corresponding each of the plurality of allocated memory blocks.